



What's New in ML?

New Features in Trilinos 7.0

Chris Siefert (1414)

Michael Gee (TU München), Jonathan Hu (1414)
Marzio Sala (ETH Zürich) and Ray Tuminaro (1414)

Sandia National Laboratories



Outline

- Matrix-Free ML.
- MLMEX: ML for Matlab.
- MLMEX Demo.



Matrix-Free ML

- Motivation: Matrix is never assembled, but matvecs are available.
- Example: Finite element matrices

$$K_0 = \sum_e T_e K_e T_e^T$$

- Appropriate for voxel conversion problems (each element is identical) \Rightarrow heavily reduced storage.
- Few preconditioners available (e.g. Jacobi, polynomial, element-by-element).
- What's new: An AMG preconditioner for matrix-free problems.



Underlying Assumptions

$$K_0 u_0 = f_0$$

- K_0 is SPD.
- Matvecs are cheap, extracting entries is expensive.
- Graph G_0 of matrix K_0 is cheaply available.
- First coarse problem (K_1) can be stored explicitly
⇒ Standard AMG can be employed on coarse grids.



How it Works (Algorithm)

$$K_0 u_0 = f_0$$

- Use G_0 to aggregate.
- Define prolongator (P_0) in the usual way.
- Form K_1 using graph coloring to minimize # matvecs.
 - Build graph G_1 of matrix K_1 (use G_0 and P_0).
 - Color G_1 using distance-2 coloring.
 - Do matvecs with block-columns of P_0 that have the same color.
- Uses matrix-free smoother (Chebyshev) on fine grid.



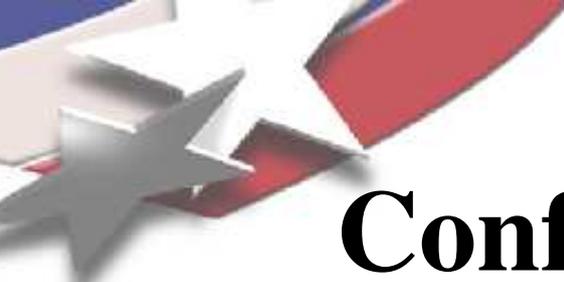
How it Works (Code)

- Required Packages: Amesos, Epetra, EpetraExt, Ifpack and Teuchos.
- Arguments to constructor
 - `const Epetra_Operator& Operator` - Matrix-free operator (K_0).
 - `const Epetra_CrsGraph& Graph` - Graph (G_0).
 - `Epetra_MultiVector& NullSpace` - Null space for ML.
 - `const Epetra_Vector& PointDiagonal` - Diagonal of K_0 .
 - `Teuchos::ParameterList& List` - Options.
- Want to see it used? Check out:
`examples/MatrixFree/ml_1D.cpp`.



What is MLMEX?

- Motivation: Fast iterative solver in Matlab
⇒ Rapid prototyping, esp. for apps using ML.
- Provides access to serial ML in Matlab.
- Supports ML_Epetra + MLAPI interfaces.
- Matlab arguments mapped to Teuchos::ParameterList
⇒ accepts any options supported by above interfaces.
- Platforms: Linux (IA-32/IA-64).



Configuring MLMEX

- Required Packages: Amesos, AztecOO, Epetra, EpetraExt, Galeri, Ifpack and Teuchos.
- New Options
 - `--enable-ml-matlab`
Turn on MLMEX compilation.
 - `--with-matlab-exec=<dir>`
Directory w/ Matlab binaries.
 - `--with-matlab-root=<root>`
Directory of Matlab install.
- For IA-64, add to configure:
`CFLAGS=-fPIC CXXFLAGS=-fPIC FFLAGS=-fPIC`



How does MLMEX work?

- Problem setup:
`[h, oc]=ml('setup', A, ...)`
- Solve problems:
`x=ml(h, A, ...)`.
- Cleanup:
`ml('cleanup', [h]).`
- Status queries :
`ml('status', [h]).`
- Interface for ML's aggregation routines:
`ml('aggregate', A, ...)`.



Setup Mode

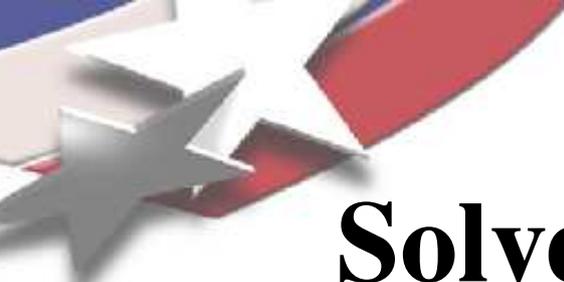
```
>> [h,oc]=ml('setup',A,['parameter',value,...])
```

- Special Options

- `mlmex: interface` (string) - Use `ML_Epetra` ('epetra') or `MLAPI` ('mlapi'). Default: 'epetra'.

- Return Values

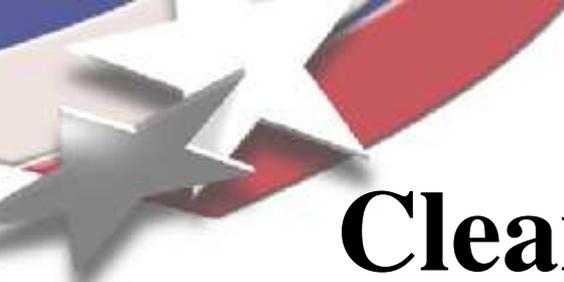
- `h` - Problem handle.
- `oc` - Operator complexity.



Solve Mode

```
>> x=ml(h,A,b,['parameter',value,...])
```

- Options such as number of Krylov iterations are specified in the MLAPI style.
 - `krylov: max iterations (int)`.
 - `krylov: tolerance (double)`.
 - `krylov: type (string) - 'cg', 'gmres' or 'fixed point'`.
 - `krylov: output level (int)`.
 - `krylov: conv (string) - 'r0', 'rhs', etc.`



Cleanup + Status Modes

```
>> ml('cleanup', [h])
```

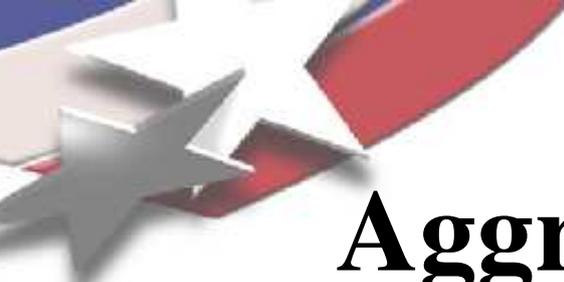
```
>> ml('status', [h])
```

- Optional Parameter: Problem handle `h`.
- If no parameter, all problems are cleaned up / printed out.



Sample Status Output

```
>> ml('status',h)
**** Problem ID 1 [ML_Epetra] ****
Matrix: 400x400 w/ 1920 nnz
Operator complexity = 1.279167e+00
Parameter List:
aggregation: threshold = 0.01    [unused]
max levels = 2    [unused]
mlmex: interface = epetra    [default]
output = 1    [unused]
```



Aggregation Mode

```
>> agg=ml('aggregate',A,['parameter',value,...])
```

- Returns a vector assigning each unknown to an aggregate.
- Cleans up used memory.
- Does not create a problem handle.